

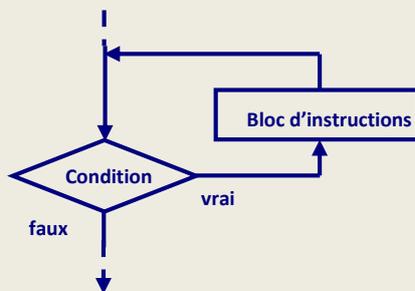


## ISN – Informatique et Sciences du Numérique

# LANGAGE PYTHON : STRUCTURES REPETITIVES

Une **structure répétitive** ou **boucle** permet de **répéter une portion de code**.

## 1 – BOUCLE WHILE (TANT QUE)



```
while Condition :  
    Bloc d'Instructions  
Suite du programme
```

**Tant que** la condition est **vraie (True)** le bloc d'instructions est exécuté. Le cycle continue jusqu'à ce que la condition soit fautive (**False**) : on passe alors à la suite du programme.

### Exemple 1 : table de multiplication par 8 avec la boucle while

```
print("Table de multiplication par 8")  
compteur = 1 # initialisation de la variable de comptage  
while compteur <= 10 :  
    # ce bloc est exécuté tant que la condition (compteur<=10) est vraie  
    print(compteur, "* 8 =", compteur*8)  
    compteur += 1 # incrémentation du compteur : compteur = compteur + 1  
  
# on sort de la boucle  
print("Eh voilà !")  
  
>>>  
Table de multiplication par 8  
1 * 8 = 8  
2 * 8 = 16  
3 * 8 = 24  
4 * 8 = 32  
5 * 8 = 40  
6 * 8 = 48  
7 * 8 = 56  
8 * 8 = 64  
9 * 8 = 72  
10 * 8 = 80  
Eh voilà !
```



### Exemple 2 : affichage de l'heure courante avec la boucle while

```
import time          # importation du module time
quitter = 'n'       # initialisation de la réponse
while quitter != 'o' :
    # ce bloc est exécuté tant que la condition (quitter != 'o') est vraie
    print("Heure courante",time.strftime('%H:%M:%S'))
    quitter = input("Voulez-vous quitter le programme (o/n) ?")

# on sort de la boucle
print("A bientôt")

>>>
Heure courante 13:56:25
Heure courante 13:56:30
Heure courante 13:56:33
A bientôt
```

## 2 – BOUCLE FOR

```
for élément in séquence :
    Bloc d'Instructions
Suite du programme
```

La **séquence est parcourue élément par élément**. L'élément peut être de tout type : entier, caractère, élément d'une liste...

L'utilisation de la **boucle for** est **intéressante** si le nombre de boucles à effectuer est **connu à l'avance**.

### Exemple 3 : table de multiplication par 9 avec la boucle for

```
print("Table de multiplication par 9")
for compteur in range(1,10) :
    print(compteur,"* 9 =",compteur*9)

# on sort de la boucle
print("Et voilà !")
```

La **valeur initiale** de l'élément compteur est égale à **1**. On **exécute la boucle tant que** l'élément compteur est **inférieur à 10**.

```
>>>
Table de multiplication par 9
1 * 9 = 9
2 * 9 = 18
3 * 9 = 27
4 * 9 = 36
5 * 9 = 45
6 * 9 = 54
7 * 9 = 63
8 * 9 = 72
9 * 9 = 81
Et voilà !
```



#### Exemple 4 : parcourt d'une chaîne de caractères avec une boucle for

```
chaîne = "Python"
for lettre in chaîne :      # Lettre est la variable d'itération
    print(lettre)
# on sort de la boucle
print("Fin de la boucle")
```

La variable **lettre** est **initialisée avec le premier élément de la séquence ('l')**. Le bloc d'instructions est alors exécuté. Puis la variable **lettre** est **mise à jour avec le second élément** de la séquence ('n') et le bloc d'instructions à nouveau exécuté... La **boucle est exécutée jusqu'à ce on arrive au dernier élément** de la séquence ('e').

```
>>>
P
y
t
h
o
n
Fin de la boucle
```

#### Exemple 5 : parcourt d'une liste avec une boucle for

```
liste = ["Pierre", "Dupont", 67.5, 17]
for element in liste :      # element est la variable d'itération
    print(element)
# on sort de la boucle
print("Fin de la boucle")
```

La variable **liste** est **initialisée avec le premier élément de la séquence ('Pierre')**. La boucle est exécutée jusqu'à ce on arrive au dernier élément de la séquence ('17').

```
>>>
Pierre
Dupont

67.5
17
Fin de la boucle
```



### 3 – INSTRUCTION BREAK

L'instruction **break** provoque une sortie immédiate d'une boucle **while** ou d'une boucle **for**.

#### Exemple 6 : instruction break

```
import time          # importation du module time
while True :        # l'expression est toujours vraie
    print("Heure courante",time.strftime('%H:%M:%S'))
    quitter = input("Voulez-vous quitter le programme (o/n) ?")
    if quitter = 'o' :
        break

# on sort de la boucle
print("A bientôt")
```

L'expression **True est toujours vraie** : il s'agit d'une **boucle sans fin**. L'instruction break est donc le seul moyen de sortir de la boucle.

```
>>>
Heure courante 09:04:02
A bientôt
```